Alignment

Plain TeX has macros to centre, left align and right align text:

```
\def\line{\hbox to\hsize}
\def\leftline#1{\line{#1\hss}}
\def\rightline#1{\line{\hss#1}}
\def\centerline#1{\line{\hss#1\hss}}
```

These macros centre, or left align or right align a *line*.

(If you try to set a line of text that is wider than the measure with \centerline it will run it off the edges of the page.) Here is a multiline equivalent of \centerline that I found on the internet:

```
\def\centre{\parindent0pt\leftskip=0pt plus 1fil
\rightskip=0pt plus 1fil
\parfillskip=0pt\parskip=0pt\obeylines}
```

It could be used in a heading macro if the headings are quite long, or where the measure is narrow, e.g. in multi-column setting:

```
\def\heading#1{\goodbreak\bigskip
{\centre\Rm\spaceskip.4em#1\par}
\nobreak\smallskip\nobreak\vp}
```

The purpose of \goodbreak before the text and \nobreak after it is to avoid the situation where a heading comes by itself at the bottom of a page with its text following on the next page. (It is saying 'please do a page break before the heading, not after it'.) The \vp macro suppresses the paragraph indent immediately after the heading.

I've also found a more concise centering macro on page 34 of *Einführung in TeX*:

```
\def\zentrierung{\obeylines\everypar{\hfil}
\parindent=0pt}
```

\obeylines makes each line of input a paragraph. \parindent= 0pt removes the paragraph indent. \everypar{\hfil} means that every paragraph-line will have horizontal fill pushing in from

\centerline, centering macros

Side notes in this document are set with the \marginalia macro

\obeylines

the left. But by default every paragraph already has `\parfill-skip=0pt plus 1fil` to fill up the last line of the paragraph. So every one line paragraph set by this macro has horizontal fill to the left and to the right of it, with the result that the lines are centered. If you were to put `\parfillskip=0pt` in this macro the lines would be right justified. (Which would be the same as using `{\obeylines\everypar{\hfill}... \hfill` is a stronger horizontal fill than `\hfil`.)

<div align="center">

Gott segne

Kupfer, Druck und

jedes andere

vervielfältigende Mittel,

so daß das Gute,

was einmal da war,

nicht wieder

zu Grunde gehen kann.

Johann Wolfgang Goethe

</div>

`\raggedright` For left aligning text longer than a line Plain has the `\raggedright` macro:

```
\def\raggedright{\rightskip\z@ plus2em
\spaceskip.3333em \xspaceskip.5em\relax}
```

`\z@` is a Plain TeX abbreviation for 0 or 0pt.

`\raggedleft` There is also a `\raggedleft` (from the `\marginalia` macro):

```
\def\raggedleft{\spaceskip=.3333em \xspaceskip=.5em
\parfillskip=0pt \leftskip=0pt plus\hsize}
```

The `\obeylines` command used as part of the centering macros above can also be used to set verse. I centered the longest line of this poem with `\centerline`, then I used `\leftskip` to push the whole poem rightwards so that it is centered on that longest line. `\parindent` is set to `0pt` to remove the paragraph indent.

<div align="center">

The Second Coming

</div>

Turning and turning in the widening gyre

The falcon cannot hear the falconer;

Things fall apart; the centre cannot hold;

Mere anarchy is loosed upon the world,

The blood-dimmed tide is loosed, and everywhere

The ceremony of innocence is drowned;

The best lack all conviction, while the worst
Are full of passionate intensity.

Surely some revelation is at hand;
Surely the Second Coming is at hand.
The Second Coming! Hardly are those words out
When a vast image out of Spiritus Mundi
Troubles my sight: somewhere in sands of the desert
A shape with lion body and the head of a man,
A gaze blank and pitiless as the sun,
Is moving its slow thighs, while all about it
Reel shadows of the indignant desert birds.
The darkness drops again; but now I know
That twenty centuries of stony sleep
Were vexed to nightmare by a rocking cradle,
And what rough beast, its hour come round at last,
Slouches towards Bethlehem to be born?

W. B. Yeats

longest line:
Troubles my sight: somewhere in sands of the desert

Somewhat similarly to \obeylines, the \obeyspaces command makes TEX take account of each indvidual space character. Normally TEX, like HTML, collapses more than one space character down into one:*   `\obeyspaces`

I ' m   a l l   s p a c e d   o u t.
I'm alright really.

For more complicated kinds of alignment TEX has the \settabs command and the \halign and \valign commands   `\settabs`

Here is an example of an alignment made with \settabs:

one	two	three	four
eins	zwei	drei	vier

First you issue a \settabs command that specifies how TEX should divide the available horizontal space into columns. \settabs 4\columns divides the \hsize into four equal columns. Then

* This applies to the ASCII space U+0020, and No-break Space U+00A0. There are other spaces in Unicode to which this does not apply.

each row of the actual table consists of the \+ control sequence followed by a list of entries separated by ampersands. The end of a row is indicated by \cr.

The other way of using \settabs is to write a template line, putting the widest entry for each column of the table in that position of the template, e.g. for this table the first line has the widest entries, so it is also used as template.

sorting algorithm	100 elements	200 elements	500 elements
bubble sort	250	1000	10000
insertion sort	200	400	3000
selection sort	110	260	2000
shell sort	70	250	700
heap sort	50	100	300
quicksort	40	60	200

If you want more control over how the columns or rows of a table appear and vertical or horizontal rules you need \halign or \valign.

\halign is kind of similar to \settabs but the template line gives you much more control over how the table is typeset. The # is used to represent the contents of the columns, and the & the divisions between the columns. If you want a column to be in a particular font it's easier to put the font command in the template. You can also use \hfil to align the text of the columns –

#\hfil	Text is left aligned
\hfil#	Text is right aligned
\hfil#\hfil	Text is centered

The columns automatically expand to the width of the widest entry e.g.
```
\halign{\it#\hfil&#&#&\bf#&#&#&\hfil\tt#\hfil\cr
   abc & d & e & f & g & h & i\cr
   a & b & c & d & e & f & ghj\cr}
```
produces:

abc d e **f** g h   i
a   b c **d** e f ghj

Space before an ampersand in the template is taken account of, but space after it seems not to be:

4

```
\halign{# &#&#&## #\cr
a&b&c&d&e\cr}
```
makes

a bcde

So if you find a really long template hard to read, you can divide
it up cell by cell by pressing carriage return after the ampersands.

Deleting the spaces in front of the ampersand in *all* the rows
of the table also seems to get rid of spaces between the affected
columns, but generally you want there to be spaces between the
columns. Usually when I create a table I put spaces on either side
of & and then use the tab key to even out the &s, to make the
input easier to read.

If you know in advance how many columns and rows your table
will have, you can write the template and space the amperands in
the rows with the tab key:

```
\halign{#&#&#&#&#&#&#\cr
  & & & & & & \cr
  & & & & & & \cr}
```

(If you run TEX now it will just produce an empty \halign.) Now
you can can set the template for each column and fill in the blanks
in the table:

```
\halign{\it#\hfil&#&##\bf#&#&##\hfil\tt#\hfil\cr
abc & d & e & f & g & h & i \cr
a   & b & c & d & e & f & ghj \cr}
```

This is slightly easier to read than the first attempt.

If you have an alignment which is made up of the same template
repeated an unspecified number of times you can use && to repeat
the template information as often as it is needed:

```
\halign{\it#&&\it#\cr
a&b&c&d&e&f&g&h&i&j&k&l&m&n&o&...\cr}
```
makes

abcdefghijklmno...

The \tabskip command allows you to set the glue between   \tabskip
columns of a \halign (or rows of a \valign).

If you set \tabskip globally outside of a \halign there will
be the same space between all the columns of the table and to
the left and right of the table. In the first line of letters below,
\tabskip0pt plus 1fil has been set this way. This is equivalent

to having a `\hfil` between the columns and on either side of the alignment. In the second line, `\tabskip0pt plus 1fil` has been set within the `\halign` and only takes effect after the first column. This and `\tabskip0pt` set in the last column, means that this alignment has no space to the left and right of it:

	a		b		c		d		e		f		g		h		i	
a		b		c		d		e		f		g		h		i

`\halign to\hsize` creates an alignment that fills the entire width of the text block.

Some examples of ruled tables: `\halign`s (i–iv) and a `\valign` (v).

(i)

a	b	c
d	e	f
g	h	i

(ii)

a	b	c
d	e	f
g	h	i

(iii)

a	b	c
d	e	f
g	h	i

(iv)

a	b	c
d	e	f
g	h	i

(v)

a	b	c
d	e	f
g	h	i

In a `\halign` continuous horizontal rules are created by placing `\noalign{\hrule}` above and below the table and between the rows of the table. Vertical rules are created row by row by putting `\vrule` in the template. You also need to have a `\strut` in the template to space the rows properly and to make the `\vrule`s continuous.

When I first started doing ruled tables I made the template like in (i) and (ii), putting the `\vrule`s in the same cells as the text:

```
\halign{\strut\vrule#&\vrule#&\vrule#\vrule\cr
```

This works fine if there are only continuous horizontal lines in the table, but if for some reason you need non-continuous horizontal lines, e.g. in (ii), it does not work very well. The only way to avoid the nicks in the second and fourth vertical rule in (ii) is to write the table so that each `\vrule` has its own cell:

```
\halign{\strut
\vrule#&#&\vrule#&#&\vrule#&#&#\vrule\cr
```

In the method used in (i) and (ii) I counted the columns and for every n columns there were n #s and $n-1$ &s in each input line (the same as for an unruled table). In the method used in (iii) and (iv) I counted the vertical rules and for every n vertical rules there were $2n-2$ &s and $2n-1$ #s. Or, if you count the columns with text in them, for n columns there are $2n$ &s and $2n+1$ #s. Here is a pseudocode input line showing were the vertical rules go:

```
|& &|& &|& &|\cr
```

Another advantage of giving each `\vrule` its own cell is that if you use `\tabskip` to space out the cells it is applied on both sides of the `\vrule`:

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

If you put the `\vrule` in the same cell as the text it is not:

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

The lines with the non-continuous `\hrule`s in them are kind of hard to figure out. They have an `\omit` in them and `\multispan` and `\hrulefill` to get a `\hrule` over multiple cells. `\offinterlineskip` being in force means that they only take up the `height` of a `\hrule` (0.4pt by default). I started with a normal input line and modified it to suit:

```
&  &&  &&  &\cr
\omit\vrule& &\multispan3\hrulefill& &\cr
```

I followed the example given in *Einführung in TEX*, pp. 106–107. See also *TEX by Topic*, 25.4.4.

If you want non-continuous vertical lines in a `\halign`, all you have to do is put `\omit` in the cell where you don't want a `\vrule`.

`\valign` is a bit like `\halign` turned sideways: the lines terminated by `\cr` go down the way instead of across. You have to use `\hsize` to specify the width of the columns, otherwise the first column will likely take up the entire measure. You can do that for the whole table like this:

```
{\hsize1.2em\parindent0pt
\valign{}
}
```

In the `\valign` the vertical rules are continuous, created by putting `\noalign{\vrule}` in the same places as `\noalign{\hrule}` in the `\halign`, and the horizontal lines are created column by column by putting `\hrule`s in the template.

In the \valign you also need to put a \strut in each of the cells of the template to make the letters align properly.

In the \haligns I used the \hbox to1.2em{} construction to equalize the widths of the \haligns and the \valign.

A table with non-continuous horizontal and vertical lines, from *The Loom of Language*:

	MASC. SING.	NEUTER SING.	FEMIN. SING.	PLURAL	MASC.	NEUTER	FEMIN.
Nomin.	DER	DAS	DIE		EIN		EINE
Acc.	DEN				EINEN		
Gen.	DES		DER		EINES		EINER
Dat.	DEM		DEN		EINEM		

A table of the 16 possible truth functions of two binary variables (from Wikipedia):

p	q	F^0	NOR^1	$\nleftarrow^2$	$\neg p^3$	$\nrightarrow^4$	$\neg q^5$	XOR^6	$NAND^7$
T	T	F	F	F	F	F	F	F	F
T	F	F	F	F	F	T	T	T	T
F	T	F	F	T	T	F	F	T	T
F	F	F	T	F	T	F	T	F	T
Com	✓	✓						✓	✓
L id			F					F	
R id					F			F	

p	q	AND^8	$XNOR^9$	q^{10}	$\rightarrow^{11}$	p^{12}	$\leftarrow^{13}$	OR^{14}	T^{15}
T	T	T	T	T	T	T	T	T	T
T	F	F	F	F	F	T	T	T	T
F	T	F	F	T	T	F	F	T	T
F	F	F	T	F	T	F	T	F	T
Com	✓	✓						✓	✓
L id	T	T	T,F	T				F	
R id	T	T			T,F	T		F	

The superscript numbers in the first line of the table refer to the first column of the wide table below.

I used `\setbox\strutbox=\hbox{\vrule height10.5pt depth3.5pt width0pt}` to space out the rows of these tables. This makes `\strut` in these tables slightly taller than it would normally be (`height=8.5pt` is the default Plain TeX setting). Roughly the same effect could be achieved by putting `\noalign{\vskip2pt}` between the rows of the table. This would work fine in a table without vertical rules, but causes gaps between the `\vrules` in tables with vertical rules. Extending the `\struts` ensures that there are no gaps in the vertical rules.

			operator	Operation name
0	(F F F F)(p, q)	⊥	false, Opq	Contradiction
1	(F F F T)(p, q)	NOR	p ↓ q, Xpq	Logical NOR
2	(F F T F)(p, q)	↚	p ↚ q, Mpq	Converse nonimplication
3	(F F T T)(p, q)	¬p, ~p	¬p, Np, Fpq	Negation
4	(F T F F)(p, q)	↛	p ↛ q, Lpq	Material nonimplication
5	(F T F T)(p, q)	¬q, ~q	¬q, Nq, Gpq	Negation
6	(F T T F)(p, q)	XOR	p ⊕ q, Jpq	Exclusive disjunction
7	(F T T T)(p, q)	NAND	p ↑ q, Dpq	Logical NAND
8	(T F F F)(p, q)	AND	p ∧ q, Kpq	Logical conjunction
9	(T F F T)(p, q)	XNOR	p if and only if q, Epq	Logical biconditional
10	(T F T F)(p, q)	q	q, Hpq	Projection function
11	(T F T T)(p, q)	p → q	if p then q, Cpq	Material implication
12	(T T F F)(p, q)	p	p, Ipq	Projection function
13	(T T F T)(p, q)	p ← q	p if q, Bpq	Converse implication
14	(T T T F)(p, q)	OR	p ∨ q, Apq	Logical disjunction
15	(T T T T)(p, q)	⊤	true, Vpq	Tautology

The second column of this table is just the numbers in the first column converted to binary, with 'T' standing for '1' and 'F' standing for '0', e.g. no. 8, Logical AND: TFFF = 1000, $1000_2 = 8_{10}$.

Another table, showing the use of `\multispan` to put a heading above a table:

Truth table for most commonly used logical operators

P	Q	$P \wedge Q$	$P \vee Q$	$P \underline{\vee} Q$	$P \underline{\wedge} Q$	$P \Rightarrow Q$	$P \Leftarrow Q$	$P \Leftrightarrow Q$
T	T	T	T	F	T	T	T	T
T	F	F	T	T	F	F	T	F
F	T	F	T	T	F	T	F	F
F	F	F	F	F	T	T	T	T

where:

T   true
F   false
$\wedge$ AND (logical conjunction)
$\vee$ OR (logical disjunction)
$\underline{\vee}$ XOR (exclusive or)

$\underline{\wedge}$ XNOR (exclusive nor)
$\Rightarrow$ conditional "if-then"
$\Leftarrow$ conditional "then-if"
$\Leftrightarrow$ biconditional "if-and-only-if".

A more modern kind of table (not so many lines, colour):

		🍴 Sun Only	🍴 Sun Only	🍴 Sun Only	🍴 Sun Only	🍴 Sun Only
BELFAST Lanyon Place	Dep	09.00	11.05	13.05	16.05	19.05
Lisburn	Dep	09.13	. .	. .	. .	. .
Lurgan	Dep	09.32	. . . .	. . . .	. . . .	. . . .
Portadown	Dep	09.40	11.36	13.36	16.36	19.36
Newry	Dep	10.01	11.58	13.58	16.58	19.58
DUNDALK Clarke	Dep	10.19	12.17	14.17	17.17	20.17
DROGHEDA MacBride	Dep	10.40	12.40	14.40	17.40	20.40
DUBLIN Connolly	Arr	11.20	13.15	15.15	18.15	21.15

table commands

A table from David Bausum's TeX page giving the primitive TeX commands relating to tables. I set the table headings O'Reilly-style in white type on a black backround.

CtrlSeq	Type	Description
\cr	c	is a visible command which ends one row in a table.
\crcr	c	is an alternate to \cr.
\everycr	pt	holds tokens inserted after every \cr or nonredundent \crcr.
\halign	c	begins the horizontal alignment of material (i.e., makes a table containing rows).
\noalign	c	inserts vertical mode material after a \cr in a table.
\omit	c	is used in the body of a table to change an entry's template from the one in the preamble.
\span	c	combines adjacent entries in a table into a single entry.
\tabskip	pg	is optional glue put between columns in a table.
\valign	c	begins the vertical alignment of material (i.e., makes a table containing columns).

In this table, c = command, pg = Parameter (glue), pt = Parameter (token).

This table shows the use of \vtop to set entries that are too wide to fit in a single row of a column. Putting them inside a \vtop allows them to be set as short paragraphs. I used \noalign{\vskip2pt} here instead of the enlarged \strut that I used in the truth tables because I only wanted to add space between the entries. Using the \strutbox method would also have spaced out the lines of the \vtops.

Actually I could have done it equivalently by setting \setbox\strutbox=\hbox{\vrule height10.5pt depth3.5pt width0pt} globally in the table and putting the normal Plain TEX \strut inside the braces of the \vtop. This is how I did it in this second table of Plain TEX table commands (definitions from *TEX for the Impatient*).

CtrlSeq	Description
`\hidewidth`	ignore width of an entry in an alignment, so that it extends out from its box in the direction of the `\hidewidth`.
`\multispan`	make next alignment entry span a specified number of columns (or rows).
`\offinterlineskip`	inhibit interline glue from now on. (Used in alignments in conjunction with `\strut`.)
`\openup`	increase `\baselineskip`, `\lineskip`, and `\lineskiplimit` by a specified amount
`\strut`	box with zero width, but height and depth of a standard line, from baseline to baseline, in the current font.

`\dotfill`, `\hrulefill` etc.

TeX has various 'fill' commands to fill up a horizontal space with a pattern or rule, e.g. `\dotfill` and `\hrulefill`:

1 2

1―――――――――――――――――――――――――2

`\leftarrowfill` and `\rightarrowfill`:

|⟵――――――――――― \hsize ―――――――――――⟶|

`\downbracefill` and `\upbracefill`:

`\leaders`

There are also `\leaders`, `\cleaders` and `\xleaders` which fill a horizontal or vertical space with copies of a pattern. These commands can be used in the typesetting of old-fashioned contents pages, e.g. in the `\halign` below `\leaders\hbox{ . }\hfil` fills the space between the second and third columns with repeating dots:

With `\cleaders` the leaders are centered within the enclosing box (at the expense of vertical alignment):

`\xleaders` seems to try and spread out the leaders as much as possible:

In this example I used a macro to create a two dot pattern:

A table of pronouns showing the use of `\upbracefill`:

Sing.	Nom.	*I*	*thou, you*	*who*	*what*
	Obj.	*me*	*thee, you*	*whom*	*what*
Plur.	Nom.	*we*	*ye, you*		
	Obj.	*us*	*ye, you*		
Sing.	Nom.	*he*	*it*	*she*	
	Obj.	*him*	*it*	*her*	
Plur.	Nom.		*they*		
	Obj.		*them ('em)*		

Some alignments showing the use of TEX math mode together with `\halign` to create tables with large braces. E.g. to create this:

$$\left\{\begin{matrix} \text{one} \\ \text{two} \\ \text{three} \\ \text{four} \end{matrix}\right\}$$

make up the `\halign` first and then paste it over the 'x' in this:
`$\left\{\matrix{\vtop{x}}\right\}$`
Placing the `\halign` inside a `\vtop{}` makes it possible to have a `\halign` within math mode and also excerpts it from TEX's math typesetting rules. If you don't want braces on the left side or the right side use `\left.` or `\right.` (You can't just leave out `\left` or `\right`.) Also the `\left` can be a right brace: `\left\}` and vice versa.

Some examples:

	PRESENT TENSE			IMPERFECT *or* PAST	
I call	*or* I am		I called	*or* I was	
thou callest	*or* thou art		thou calledst	*or* thou wast	
he calls	*or* he is	} calling	he called	*or* he was	} calling
we call	*or* we are		we called	*or* we were	
you call	*or* you are		you called	*or* you were	
they call	*or* they are		they called	*or* they were	

	FUTURE		
I shall		I shall be	
thou wilt		thou wilt be	
he will	} call *or* {	he will be	} calling
we shall		we shall be	
you will		you will be	
they will		they will be	

The `\halign`s that are on the same level horizontally are stacked over each other in the input, with no blank lines between them e.g. the 'Present Tense' and 'Imperfect' tables look like this with the `\halign`s taken out. The `\halign`s go over 'x' and 'y':

```
\noindent
$\left.\matrix{\vtop{x}}\right\}$ calling\quad
$\left.\matrix{\vtop{y}}\right\}$ calling
```

14

(Below) an example of using big braces within a \halign. I made up the table first without the braces, then wrote macros for the braces themselves, to try and make the input a little easier to follow.

I've rewritten the macros for the braces. The original way involved too much rewriting of the table. The new macros can be just 'dropped in'. You can use \kern to adjust their horizontal position. The macros use \smash and \lower to fit the braces into the table without disturbing the linespacing. The \smlower macro is used to \smash and \lower the words in the cell to the right to align them with the points of the braces.

ANGLO-AMERICAN	BIBLE ENGLISH	OLD ENGLISH	GERMAN
I do	I do	ic do	ich tue
you	thou doest	thu dest	du tust
he does	he doeth	he deth	er tut
we	we	we	wir tun
you do	you do	ge doth	ihr tut
they	they	hie	sie tun
I did	I did	ic dyde	ich tat
you	thou didst	thu dydest	du tat(e)st
he	he did	he dyde	er tat
we	we	we	wir taten
you	you	ge dydon	ihr tatet
they	they	hie	sie taten
I have done	I have done	ic haebbe gedon	ich habe getan
I had done	I had done	ic haefde gedon	ich hatte getan
(to) do	(to) do	don	(zu) tun

ANGLO-AMERICAN	BIBLE ENGLISH	OLD ENGLISH	GERMAN
I, you } do he does	I do thou doest he doeth	ic do thu dest he deth	ich tue du tust er tut
we, you, they } do	we, you, they } do	we, ge, hie } doth	wir tun ihr tut sie tun
I, you, he, we, you, they } did	I did thou didst he we, you, they } did	ic dyde thu dydest he dyde we, ge, hie } dydon	ich tat du tat(e)st er tat wir taten ihr tatet sie taten
I have done	I have done	ic haebbe gedon	ich habe getan
I had done	I had done	ic haefde gedon	ich hatte getan
(to) do	(to) do	don	(zu) tun

Another table, slightly trickier than the previous one.

PERSONAL PRONOUNS

UNSTRESSED   STRESSED

SINGULAR

Persons	Nom. Subj.	Accus.	Dat.	Genit.	Nom. and Accus.
1st	je	me	me		moi
2nd	tu	te	te		toi
3rd masc.	il	le	lui, y	en	lui
3rd fem.	elle	la			elle
3rd neut.	il	le	y		
3rd reflex.		se	se		soi

PLURAL

Persons	Nom. Subj.	Accus.	Dat.	Genit.	Nom. and Accus.
1st	nous	nous	nous		nous
2nd	vous	vous	vous		vous
3rd masc.	ils	les	leur, y	en	eux
3rd fem.	elles				elles
3rd reflex.		se	se		soi

16

Some examples (from *Einführung in TEX*, p. 39) showing various settings of `\hangindent` and `\hangafter`. The default settings are `\hangindent=0pt` and `\hangafter=1`.

`\hangindent=2cm\hangafter=-4\noindent`

          Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

`\hangindent=2cm\hangafter=4\noindent`

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

`\hangindent=-2cm\hangafter=-4\noindent`

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

`\hangindent=-2cm\hangafter=4\noindent`

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupi-

datat non proident, sunt in culpa qui officia deserunt
mollit anim id est laborum.

For a normal hang paragraph you would use the Plain TEX
hang command:
```
\def\hang{\hangindent\parindent}
```
in conjunction with `\noindent`:

```
\hang\noindent
```
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
   eiusmod tempor incididunt ut labore et dolore magna aliqua.

`\hangindent` and `\hangafter` are reset to their default values
after each paragraph. If you have a lot of text set in hang
paragraphs, and you don't feel like placing `\hang\noindent` before
each paragraph, put this around your paragraphs:
```
{\everypar{\setbox0=\lastbox\hang}
First paragraph.

Second paragraph.

}
```
e.g.
Ampersand: The name given to the contraction of 'and', thus: &.
Antiqua: The German name for roman type.
Antique paper: A term used to describe any good book paper with
   a rough surface.
Art paper: A clay-coated paper, especially suitable for half-tone
block printing. This is generally produced with a highly glazed
surface but when the surface is matt it is known as matt art
paper.

But note that hang paragraphing has failed in the fourth
paragraph because I put the first two words in `{\it }`. There is a
workaround: either put `\leavevmode` before `{\it Art paper:}`,
or put `\global` in `\everypar`:
```
\everypar{\global\hang\setbox0=\lastbox}.
```
See here. Another way of doing it is to use something similar to the
`\entry` macro which I used in Glossary63.tex on the Typography
page:

Ampersand: The name given to the contraction of 'and', thus: &.

Antiqua: The German name for roman type.

Antique paper: A term used to describe any good book paper with a rough surface.

Another way of using `\hangindent` to set a glossary:

Ampersand: The name given to the contraction of 'and', thus: &.

Antiqua: The German name for roman type.

Antique paper: A term used to describe any good book paper with a rough surface.

This is adapted from the `\hangfrom` macro in Section 18.3.3 of *TEX by Topic.* I gave `\hangobject` a fixed value close to what I thought would be the longest head word. The end result is kind of similar to the `\halign` with `\vtop` method on p. . If the head word is unusually long you will have to break it:

Electronic scan- This is the latest method of colour separation
ning and   and engraving which can be used in conjunction
engraving: with either letterpress printing or lithography.

In the original version `\hangobject` takes the width of the head word:

Ampersand: The name given to the contraction of 'and', thus: &.

Antiqua: The German name for roman type.

Electronic scanning and engraving: This is the latest method
                              of colour separation and en-
                              graving which can be used in
                              conjunction with either let-
                              terpress printing or lithogra-
                              phy.

The original macro was probably written with short head words in mind.

A macro for producing a paragraph with a centered last line:

```
\def\centrepar{\rightskip 0pt plus -1fil
\leftskip 0pt plus 1fil
\parfillskip 0pt plus 2fil\relax}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

`\parshape` For more precise control over the shape of a paragraph, TeX has the `\parshape` command. This command has the form `\parshape` n i_1l_1 $i_2l_2 \ldots i_nl_n$ where n is the number of lines you want to control, i is the indentation and l is the remainder of the line after the indentation (which would be `\hsize` $- i$, if you want the right margin to be justified). Here is an extreme example of the use of `\parshape` from the TeX Showcase.

An example of a tapered paragraph. In this `\parshape`, for each indent, `\hsize` is diminished by $2i_n$, producing an equal indent on each side.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia de- serunt mollit anim id est laborum.

An example of using `\parshape` to set incised margin notes. I used `\parshape` to make the indentation then I reset `\marginalia`'s `\inmarginspacing` parameter to `-4em` to push the note into the paragraph.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu

A cut-in margin note

fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

A macro for a lettrine, built with the help of `\parshape`:

Secundum Lucam c. 9

I N illo témpore, Elevátis Jesus óculis in discípulos suos, dicébat; Beati páuperes: quia vestrum est regnum Dei. Beáti qui nunc esurítis: quia saturabímini. Beáti qui nunc fletis: quia ridébitis. Beáti éritis cum vos óderint hómines, et cum separáverint vos et exprobráverint et ejécerint nomen vestrum tanquam malum propter Fílium Hóminis. Gaudéte in illa die et exsultáte: ecce enim merces vestra multa est in cœlo.

Originally I wrote the macro like this:

```
\def\cap#1#2{\vskip\baselineskip
\noindent\smash{\raise0.5pt\hbox{\capfont#1}}\nobreak
\vskip-2\baselineskip\nobreak
\parshape 3 5mm 43mm 7mm 41mm 0mm 48mm\nobreak
\noindent{\scw #2}\nobreak}
```

But when the drop capital came immediately after the `\doublecolumns` command this happened:

I N illo témpore, Elevátis Jesus óculis in discípulos suos, dicébat; Beati páuperes: quia vestrum est regnum Dei. Beáti qui nunc esurítis: quia saturabímini. Beáti qui nunc fletis: quia ridébitis. Beáti éritis cum vos óderint hómines, et Secundum Lucam c. 9 cum separáverint vos et exprobráverint et ejécerint nomen vestrum tanquam malum propter Fílium Hóminis. Gaudéte in illa die et exsultáte: ecce enim merces vestra multa est in cœlo.

Replacing `\vskip\baselineskip` in the first line of the macro with `\null` fixed the problem. `\null` – `\def\null{\hbox{}}` – puts an empty `\hbox` on the first line of the paragraph, so that the drop capital now rests on the second baseline. `\noindent\smash{\kern0pt\raise0.5pt\hbox{\capfont#1}}` is the code that positions the drop capital. `\raise` (or `\lower`) can be used to move the letter vertically. You can also use `\kern` to move the letter horizontally. The purpose of `\smash` in the macro is to make TEX disregard the vertical dimensions of the drop capital. If `\smash` was not there the 'horns' of the I, or the accent on the example further on, would interfere with the linespacing. `\smash` is the opposite of the `\phantom` commands. With `\smash` the drop capital in the macro is typeset, but its `height` and `depth` are disregarded, whereas with `\vphantom` e.g. in the Plain TEX `\mathstrut` command the argument is not typeset but the `height`

and `depth` are taken account of:

   `\def\mathstrut{\vphantom{(}`

This makes an empty box with the `height` and `depth` of a left parenthesis.

   `\vskip-2\baselineskip\nobreak` brings the following text back up to the first line of the paragraph, then `\parshape` sets the indentations to make room for the drop capital, then `\noindent{\scw #2}}` puts the rest of the first word* in letterspaced small caps, to make a transition between the drop cap and the normal text.

   In the lettrine macro, if the drop capital takes up n lines, on the first line you would have `\null\vglue[n - 2]\baselineskip`,† and on the second line `\vskip-n\baselineskip`. Since n equalled 2 in the example, the first line of the macro just used `\null`.

   When I am making lettrine macros I start with a generic macro for 'normal' letters like H or N.

Having now gained some insight into the general aspects of the composing department, as well as the materials, tools, and other appliances used therein, it is necessary for the student to acquire a knowledge of the boxes of the upper and lower cases respectively.

   Then I make customized versions for some of the other letters; I and M obviously:

In connection with composition the classical or foreign languages which require special cases, as also some amount of experience in setting, must not be forgotten. The languages which a compositor has to deal with more generally are Greek, Hebrew, and German.

Music is printed by one of the four following processes: plate, lithography, photo-zinco, and type, the last method being more generally used for book music. The composition requires some amount of practice, as this class of work is almost distinct from ordinary letterpress printing. The cases used are very complicated, and contain more boxes than are usually required for

---

* or the first three or four words, or the whole first line.

† I used `\vglue` here because it produces vertical space that does not disappear at a page break. Apart from that it is the same as `\vskip`. When I used `\null\vskip2\baselineskip` in the four line lettrines and one of them chanced to come at the top of the page, the lettrine and paragraph protruded two `\baselineskip`s above the top of the text block.

other founts, foreign or otherwise, owing to the very large number of pieces used.

But also A:

ALTHOUGH wood engraving as an art is almost extinct, woodcut printing, whether on the hand-press or on power machine, is capable of a large amount of treatment. To bring out the degrees of light and shade of an illustration which has much work in it is a task which requires a great deal of experience; in fact, the workman should possess some artistic qualification to appreciate and to give effect to the artist's design.

Maybe P:

PERFECTING machines, with two cylinders, are those which print both sides of the sheet before it leaves the machine, but with two quite distinct impressions, and are adapted for ordinary bookwork of long numbers.

And also J, L, O, Q, V, W, Y. If I was setting a book with a lot of lettrines in it I might end up writing macros for most of the alphabet.

A four line drop capital. The 'Q' is raised slightly so that the tail does not hit the letters below it.

QU'EN PLEIN MARAIS de la Souteyranne, à quelques kilomètres au nord d'Aigues-Mortes, se trouve la Tour Carbonnière, surprend les visiteurs. Construite au XIIIe siècle, elle contrôlait l'unique voie d'accès terrestre de la ville fortifiée, celle qui menait à Psalmody, l'une des «abbayes de sel» dont il ne reste que quelques vestiges. L'abbaye était ravitaillée par un souterrain qui la reliait au château de Treillan.

A drop cap V, showing how the letter is optically kerned into the margin and how the text slopes around the letter.

VOICI à quelques kilomètres au nord d'Aigues-Mortes la Tour Carbonnière. Construite au XIIIe siècle, elle contrôlait l'unique voie d'accès terrestre de la ville fortifiée, celle qui menait à Psalmody, l'une des «abbayes de sel» dont il ne reste que quelques vestiges. L'abbaye était ravitaillée — dit-on — par un souterrain qui la reliait au château de Treillan.

An A with a diacritic:

À QUELQUES KILOMÈTRES au nord d'Aigues-Mortes, se trouve la Tour Carbonnière. Construite au XIIIe siècle, elle contrôlait l'unique voie d'accès terrestre de la ville fortifiée, celle qui menait à Psalmody, l'une des «abbayes de sel» dont il ne reste que quelques vestiges. L'abbaye était ravitaillée — dit-on — par un souterrain qui la reliait au château de Treillan.

A drop capital with a punctuation mark in front of it:

«EN plein marais de la Souteyranne, à quelques kilomètres au nord d'Aigues-Mortes, se trouve la Tour Carbonnière. Construite au XIIIe siècle, elle contrôlait l'unique voie d'accès terrestre de la ville fortifiée, celle qui menait à Psalmody, l'une des «abbayes de sel» . . .

A Gothic initial:

WER reitet so spät durch Nacht und Wind?
Es ist der Vater mit seinem Kind;
Er hat den Knaben wohl in dem Arm,
Er faßt ihn sicher, er hält ihn warm.

Mein Sohn, was birgst du so bang dein Gesicht? –
Siehst, Vater, du den Erlkönig nicht?
Den Erlenkönig mit Kron und Schweif? –
Mein Sohn, es ist ein Nebelstreif. –

These last four lettrines are taken from examples of the LaTeX \lettrine package.

Plain TeX by default inserts additional space after certain punctuation marks. This is set by the \nonfrenchspacing macro:

```
\def\nonfrenchspacing{\sfcode`\.3000\sfcode`\?3000%
\sfcode`\!3000\sfcode`\:2000\sfcode`\;1500%
\sfcode`\,1250 }
```

The \sfcode (*space factor code*) command in this macro is used to alter the spacing after individual characters:

Public. junk? dwarves! hug: my; quartz, fox.

(Compare with \frenchspacing:

Public. junk? dwarves! hug: my; quartz, fox.)

This macro will not add additional space if the punctuation mark follows a capital letter, because it assumes that the capital letter is an initial in someone's name: D. E. Knuth. If you want to

have the extra spacing after a punctuation mark that is preceded by caps, you could do something like this:

```
\TeX\ isn't WYSIWYG\null? 'Fraid not.
```

TEX isn't WYSIWYG? 'Fraid not.

You could write your own macro to apply the spacing after punctuation marks that you prefer:

```
\def\myspacing{\sfcode`\.2500\sfcode`\?2500%
\sfcode`\!2500\sfcode`\:1700\sfcode`\;1250%
\sfcode`\,1000 }
```

Sphinx. of? black! quartz: judge; my, vow.

If you want spacing after punctuation to be uniform use the \frenchspacing macro:

```
\def\frenchspacing{\sfcode`\.1000 \sfcode`\?1000%
\sfcode`\!1000\sfcode`\:1000 \sfcode`\;1000%
\sfcode`\,1000}
```

OK, that's spacing after punctuation marks, but what if you want to alter spacing before punctuation marks? In old books there always seems to be more spacing in front of colons, semicolons, exclamation marks and question marks than in modern digitally set books. And in other languages such as French, there seems to be a lot more spacing before these punctuation marks than in English.

I found a method of altering the spacing before punctuation marks on a website which is no longer up but the files can be had via the Internet Archive. This code is taken from Velene.pdf:

```
\catcode`\: =\active
\def :{\unskip\kern .16667em\string:}
\catcode`\«=\active
\def «{\string«\kern .16667em\ignorespaces}
\catcode`\»=\active
\def »{\unskip\kern .16667em\string»}
\catcode`\;=\active
\outer\def ;{\unskip\kern .16667em\string;}
\catcode`\!=\active
\outer\def !{\unskip\kern .16667em\string!}
\catcode`\?=\active
\def ?{\unskip\kern .16667em\string?}
\catcode`\'=\active
\def '{\string'}
```

It puts additional space before colons, semi-colons, exclamation

marks, question marks, and before a right guillemet and after a left guillemet, in accordance with French typographic usage:

« Prenez garde au chien ! », lisait-on à l'entrée des maisons romaines.

« C'est par le sang et par le fer que les États grandissent », a dit Bismarck.

A-t-il dit : « Je reviendrai » ?

Quel bonheur d'entendre : « Je vous aime ! »

A-t-il dit : « qui est ici ? »

Il a dit « Je reviendrai. »

With some fonts I use these settings to add some space before : ; ! ?

```
\catcode`\:=\active
\def :{\unskip\kern 1pt\string:}
\catcode`\;=\active
\outer\def ;{\unskip\kern 1pt\string;}
\catcode`\!=\active
\outer\def !{\unskip\kern .1em\string!}
\catcode`\?=\active
\def ?{\unskip\kern 1pt\string?}
```

The catcode method works in pdfTeX, X$_{\text{E}}$TeX and LuaTeX.

pdfTeX has its own method of adding space before characters:

```
\pdfprependkern=1% activate kerns before characters
\knbccode\font`\:=100
\knbccode\font`\;=100
\knbccode\font`\!=100
\knbccode\font`\?=100
```

The numbers are thousandths of an em. In 10pt type 100 is equivalent to 1pt. See p. 32 of the pdfTeX manual.

Another method, specific to X$_{\text{E}}$TeX, of achieving the same results is with \XeTeXcharclass and \XeTeXinterchartoks:

```
\XeTeXinterchartokenstate=1
\XeTeXcharclass `\: = 1
\XeTeXcharclass `\; = 1
\XeTeXcharclass `\! = 2
\XeTeXcharclass `\? = 2
\XeTeXinterchartoks 0 1 = {\kern 0.5pt}
\XeTeXinterchartoks 0 2 = {\kern 1pt}
```
boxing: wizards; jump! quickly?

But if the preceding word is in a different font it doesn't work:
boxing: wizards; *jump*! *quickly*?
(But if the whole thing is in italic it does:
boxing: wizards; jump! quickly?)

There is also yet another way to add space before characters with LuaTEX. See here.

\XeTeXinterchartoks can also be used to fix gaps in a font's kerning table, e.g. in EB Garamond the combination 'VÆ' does not seem to be kerned: LINGVÆ. You could fix that like this:

```
\XeTeXinterchartokenstate=1

\newXeTeXintercharclass \Vclass
\newXeTeXintercharclass \AEclass

\XeTeXcharclass`\V \Vclass
\XeTeXcharclass`\Æ \AEclass

\XeTeXinterchartoks\Vclass \AEclass = {\kern -1pt}
```
$$\text{SYNTAXIS LINGVÆ GRÆCÆ}$$

Another use for **\XeTeXinterchartoks** would be to suppress the 'Th' ligature that some fonts have on by default. E.g. in Minion Pro and some other Adobe fonts the 'Th' ligature is included in standard ligatures (`liga`). You could switch off `liga` like this:
```
\font\minnolig="MinionPro-Regular:-liga,mapping=tex-
text" at 10pt
```
but then you would lose the f ligatures. If you use **\XeTeXinterchartoks** like this –

```
\XeTeXinterchartokenstate=1

\newXeTeXintercharclass \Tclass
\newXeTeXintercharclass \hclass

\XeTeXcharclass`\T \Tclass
\XeTeXcharclass`\h \hclass

\XeTeXinterchartoks\Tclass \hclass = {\kern 0pt}
\font\min="MinionPro-Regular:mapping=tex-text"    at
10pt
{\min Th ff ffi ffl fi fl}
```
– the 'Th' ligature is suppressed but the f ligatures remain.

As well as `\sfcode` TEX also has the `\spacefactor` command. This overrides `\fontdimen2`, the normal interword space. `\spaceskip` is meant for temporarily changing the values, like removing the stretch and shrink components in `\raggedright` Sometimes I use `\spaceskip.33em` in places where text does not need to be justified and I want a slightly wider word spacing, e.g. in headings or page headers. The standard `\fontdimen` registers for text fonts are

1. slant factor for positioning of accents for slanted and italic fonts.
2. normal interword space, may be overwritten by setting `\spaceskip`.
3. is the extra interword space, which may be overwritten by the 'plus' part of `\spaceskip`. This is the extra space TEX uses for justification of paragraph line.
4. is the extra interword shrinkable space, which TEX may reduce for justification. It will be overwritten by the 'minus' part of `\spaceskip`.
5. x-height, height of lowercase letters without ascenders, e.g., 'a' 'o' or 'x'. This is just the font-specific measuring unit '1 ex'.
6. '\quad' is the font specific width of 'M', simultaneously the measuring unit '1 em'.
7. additional interword space after punctuation; suppressed by `\frenchspacing`. (A non zero value of `\xspaceskip` takes precedence over it.)

The rest of the `\fontdimen` registers (up to 22) deal with mathematical typesetting.

Each font has its own `\fontdimen` settings. If you wanted to change them you could do this:

```
\fontdimen2\font=.25em % inter-word space
\fontdimen3\font=.25em % inter-word stretchability
\fontdimen4\font=.05em % inter-word shrinkability
```

This example taken from here. See also.

microtypo-
graphy

A further refinement of spacing is the micro-typographical extensions introduced by pdfTEX and since taken up by LuaTEX and XₑTEX. The most prominent of these are font protrusion, in which characters such as ' " ! , - . : ; ? are pushed out slightly into the right margin to create a more optically straight right edge, and font expansion in which glyphs can be imperceptably widened or

narrowed to aid in justification. Currently pdfTeX and LuaTeX support both font protrusion and expansion; XƎTeX supports font protrusion only. See p. 6 of the manual for the LaTeX `microtype` package.

Some XƎTeX examples using a macro for protrusion which I found here:

0. The default: no protrusion.
We thrive in information‖thick worlds because of our marvelous and everyday capacity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeonhole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, average, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, winnow the wheat from the chaff and separate the sheep from the goats.

1. The same text with `\XeTeXprotrudechars` set to 1.
We thrive in information‖thick worlds because of our marvelous and everyday capacity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeonhole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, average, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, winnow the wheat from the chaff and separate the sheep from the goats.

When `\XeTeXprotrudechars` is set to 1, the protrusion is applied after TeX's normal paragraph breaking routines have broken the paragraph into lines. The linebreaking here is the same as in example 0.

2. With `\XeTeXprotrudechars` set to 2.
We thrive in information‖thick worlds because of our marvelous and everyday capacity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeonhole,

pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, average, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, winnow the wheat from the chaff and separate the sheep from the goats.

When `\XeTeXprotrudechars` is set to 2 the changes caused by protrusion and expansion are taken into account *while* the paragraph is being built. Consequently the line breaking in this example is different to that in examples 0 and 1.

The microtypography examples are taken from an article by Hans Hagen.